

SYSTEM AND METHOD OF USER AUTHENTICATION FOR NETWORK
COMMUNICATION THROUGH A POLICY AGENT

5

TECHNICAL FIELD OF THE INVENTION

This invention relates generally to network communication and, more particularly, to the communication between a computer on a first network with another computer on a second network through a policy agent of the first network.

10

BACKGROUND OF THE INVENTION

Computers in an organization are often linked together to form a private network so that the computers can communicate with each other and share resources. Such an internal computer network within the control of an organization is commonly referred to as an "intranet." Many intranets are composed of local area networks, although the intranets of some large organizations have grown very large and require the level of sophistication found in the Internet.

15

20

Today's network environment demands secure data communications. A major concern for private networks is the possibility of security breach through communications with external networks. As the Internet and the World Wide Web become the essential backbone for worldwide commerce and information sharing, the need for manageable and secure networks becomes more urgent than ever. A fundamental key to the implementation of a secure network is the ability to manage network access. To protect the integrity and vital data of an intranet, a network administrator has to be able to

25

0946135 44099
666077 "SECRET"

implement policies to restrict access to certain users or sources. The restrictions may be based on various parameters, such as user credentials, the source address, the destination address, time of the day, etc. To that end, a policy agent
5 standing between a private network and an external network, such as the Internet, is typically the focal point for carrying out the network access policies. A policy agent may be, for example, a network firewall that guards the intranet and hides its structure from the outside by filtering
10 communication packets or performing session-based application-level access control.

A successful implementation of access control often requires the capability of applying access policies based on user credentials, i.e., who the user is, whom the user is
15 allowed to send or receive network communication to or from, etc. The commonly used network protocols, however, typically do not lend themselves to user authentication in connection with network access. Network communication data are represented in a variety of network protocols. Each of the
20 existing network protocols serves one or more technical purposes within a network environment. Typically, under those protocols, only the source and destination addresses and ports are provided in the header of the network data, and the access policies are typically limited to those parameters. The lack
25 of user information in the communication packets makes it very difficult to implement network policies based on user credentials. It is possible, of course, to create a new

665011 "SECRET" 41099

network protocol that includes information for user authentication in the data stream. Such a solution, however, may not be preferred, as it will require almost all existing network applications to be rewritten to accommodate the new
5 protocol.

SUMMARY OF THE INVENTION

12001
65607-136450

In view of the foregoing, the present invention provides a method and system for a policy agent of a network to
10 authenticate a user that uses a client computer on the network to transmit network communication data, and to associate the data stream from the client computer with the user. When the client computer initiates a network data connection to or through the policy agent, the policy agent detects the data
15 connection and sends a challenge to the client computer. The challenge is encrypted with a private key of the policy agent. When the client computer received the challenge, it decrypts the challenge with the public key of the policy agent and prepares a message digest value, such as by a hash algorithm,
20 based on the data in the challenge and the network data sent by the user. The message digest value is then encrypted with the private key of the user and sent to the policy agent. The policy agent decrypts the received response with the public key of the user to obtain the message digest value. The
25 policy agent then calculates a digest value based on the challenge and the network data received from the client computer, and compares the calculated digest value with the

a
digest value decrypted from the response. If the two digest values match, the policy agent knows that the user has been authenticated, and that the received network data are those sent by the user. The policy agent may then apply network
5 policies based on the credentials of the authenticated user.

Additional features and advantages of the invention will be made apparent from the following detailed description of illustrative embodiments which proceeds with reference to the accompanying figures.

10

BRIEF DESCRIPTION OF THE DRAWINGS

While the appended claims set forth the features of the present invention with particularity, the invention, together with its objects and advantages, may be best understood from
15 the following detailed description taken in conjunction with the accompanying drawings of which:

Figure 1 is a block diagram generally illustrating an exemplary computer system on which the present invention resides;

20 FIG. 2 is a schematic diagram showing a client computer on a network transmitting network data through a policy agent of the network;

FIG. 3 is a schematic diagram illustrating communications
25 between the policy agent and the client computer for a user authentication process;

656011-1099

FIG. 4 is a flow chart showing an embodiment of the user authentication process for network data transmitted through the policy agent;

FIG. 5 is a schematic diagram illustrating the generation of a challenge sent by the policy agent to the client computer for user authentication;

FIG. 6 is a schematic diagram illustrating the generation of a response by the client computer; and

FIG. 7 is a schematic diagram illustrating the use of the response for authenticating the user and associating the received network data to the user.

DETAILED DESCRIPTION OF THE INVENTION

Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like.

The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be
5 located in both local and remote memory storage devices.

With reference to Fig. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer 20, including a processing unit 21, a system memory
10 22, and a system bus 23 that couples various system components including the system memory to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

15 The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal
20 computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk 60, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical
25 media.

The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a

65607-5279450

hard disk drive interface 32, a magnetic disk drive interface 33, and an optical disk drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 20. Although the exemplary environment described herein employs a hard disk 60, a removable magnetic disk 29, and a removable optical disk 31, it will be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories, read only memories, and the like may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk 60, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more applications programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and a pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 47 or other

type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, personal computers typically include other peripheral output devices, not shown, such as speakers and
5 printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The remote computer 49 may be another personal computer, a server, a router, a
10 network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network
15 (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 20 is connected to the local network 51 through a
20 network interface or adapter 53. When used in a WAN networking environment, the person computer 20 typically includes a modem 54 or other means for establishing communications over the WAN 52. The modem 54, which may be internal or external, is connected to the system bus 23 via
25 the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory

656011-5292450

5 In the description that follows, the invention will be described with reference to acts and symbolic representations of operations that are performed by one or more computers, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to
10 as being computer-executed, include the manipulation by the processing unit of the computer of electrical signals representing data in a structured form. This manipulation transforms the data or maintains it at locations in the memory system of the computer, which reconfigures or otherwise alters
15 the operation of the computer in a manner well understood by those skilled in the art. The data structures where data is maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the
20 foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that various of the acts and operation described hereinafter may also be implemented in hardware.

Referring now to FIG. 2, the present invention is directed to a system and method of authenticating a user 70 that uses a client computer 72 on a network 74 to send a network data stream 76 through a policy agent 78 of the

network. As shown in FIG. 2, the policy agent 78 is the connection point, or gateway, between the network 74 of the client computer and another network 80 on which the recipient computer 82 resides. The policy agent may be, for example, a network firewall. In a preferred embodiment, the first network 74 on which the client computer 72 resides is an intranet, and the second network 80 is an external network, such as the Internet. When the user 70 uses the client computer 72 to send network data to the recipient computer 82 on the external network, the network data is directed to the policy agent 78, which applies access policies to determine whether to pass the network data to or from the recipient computer.

In accordance with a feature of the invention, the policy agent 78 carries out a user-authentication process that authenticates the user 70 who sends the network data 76 and associates the network data received from the client computer with the authenticated user. In other words, the policy agent 78 verifies that user 70 is indeed who she claims to be, and verifies that the received network data are indeed those sent by that user. The user authentication enables the policy agent to apply access policies based on the credentials of the user. As will be described in greater detail below, the user authentication according to the invention is secure from malicious alterations, such as by a man-in-the-middle attack, user credential spoofing, or a reply attack.

666077-9E9E460

hs A2

Moreover, the authentication process is independent of the particular network protocol used for transmitting the network data. In accordance with a feature of the invention, the user authentication is performed out-of-band, i.e., the network communications for the authentication process are not part of network data stream on which the network policies are to be applied. Few network protocols commonly used for network data transmission have provisions for inclusion of user authentication information as part of the data stream. If an in-band user authentication is to be implemented, those network protocols would have to be substantially modified or entirely replaced with new protocols. Such a solution, however, may not be practical or desirable. The out-of-band user authentication in accordance with the invention avoids the need to include user authentication information in the network data stream on which access policies are to be applied. As a result, the authentication process is independent of the underlying network protocol used to send the data stream and can be used with existing network protocols. With out-of-band user authentication, however, there is a risk that the user information may be altered or incorrectly associated with the network data stream. As will be described in greater detail below, the user authentication according to the invention effectively avoids this risk by including data from the network data stream in the digital signature process used in the user authentication.

0650TT-5E19E460

Turning now to FIG. 3, the user authentication process includes the sending of a challenge 90 by the policy agent 78 to the client computer 72 that initiated the network connection 92, and the returning of a response 94 by the client computer to the policy agent. This challenge-response sequence employs a public key/private key encryption scheme for authenticating the identity of the user. The response 94 also includes information regarding the network data to enable the policy agent to confirm that the received network data are those sent by the user. In a preferred embodiment, the network data are in the form of data packets 96.

An embodiment of the authentication process is illustrated in FIG. 4. When the client computer located on the network initiates a network data connection either to or through the policy agent (step 100), the out-of-band authentication process is initiated. When the policy agent detects the new network connection (step 102), it obtains the purported identity of the user (step 104). Various ways can be used to obtain the identity of the user associated with the network connection with the client computer. For example, in a preferred embodiment, the source information in the network data may include the address and port number of the client computer that sent the data. A query may be sent to a directory service 84 (FIG. 2) that keeps track of the current configuration and usage of the computers on the network. In response, the directory service 84 returns the identity of the user currently at the source address and port number. It will

Ins A3₅

~~After the policy agent has obtained the purported~~

When the client computer 72 receives the challenge 90, it decrypts the challenge using the policy agent's public key (step 112). The client computer then generates a first message digest value based on the data from the decrypted challenge and the network data transmitted to the policy agent (STEP 114). The first message digest value is encrypted with the private key of the user to form a response to the challenge (step 116), and the response is sent to the policy agent (step 118). When the policy agent receives the response, it decrypts the response with the public key of the user to obtain the first message digest value (step 120). It then calculates a second message digest value from the challenge and the network data it received from the client computer (step 122). The policy agent then compares the digest value it calculated with the digest value decrypted

5 It is to be noted that it is not necessary for the policy agent to have received the data packets from the user when it constructs the challenge. The policy agent, however, can calculate the second digest value (step 122, Fig. 4) only after having received the data packets. If the policy agent
10 receives the data packets from the client computer prior to calculating the challenge or before the user is authenticated, it should buffer the data and not forward them to their destination until the authentication process has been completed.

25 By way of example, the construction of the challenge and
response in a preferred embodiment is described below.
Referring now to FIG. 5, in this embodiment, the underlying

protocol for the transmission of network data is the widely used TCP/IP. Each of the client computer and the policy agent is identified by an IP address, and the network connection between them is further identified by a port number of the client computer and a port number of the policy agent. The network data sent by the user at the client computer are transmitted in the form of communication packets 96 (FIG. 3). When the policy agent 78 detects the network connection from client computer 72, it obtains the name of the unauthenticated user who purportedly used the client computer to initiate the connection. The policy agent then proceeds to look up the public key PUBKu of the user of the client computer. The policy agent knows its own public key PUBKp and private key PRVKp. In addition to the user's public key PUBKu, the policy agent obtains the following information from the operating system:

IPc - the IP address of the client computer;
PORTc - the IP connection port of the client computer;
PORTp - the IP connection port of the policy agent;
T - the system time;
RND1 - a random number.

The policy agent then constructs a challenge for user authentication based on these five values. As illustrated in FIG. 5, these five values are concatenated, and the resultant value 136 is used as the input of a hash function 138 to generate a hash value H1. The hash function 138 may be, for

2025 RELEASE UNDER E.O. 14176

example, the MD5 algorithm known to those skilled in the art.

This step of generating the hash value is represented by the following expression:

$$H1 = MD5(RND1*PORTp*PORTc*IPc*T),$$

5 where the symbol "*" means concatenation. The hash value H1 is then encrypted first with the private key PRVKp of the policy agent and then with the public key PUBKu of the purported user to form the challenge 90. This step is represented by the following expression:

$$10 \quad C = PUBKu(PRVKp(H1)),$$

wherein C is the challenge. The challenge 90 is sent by the policy agent 78 to the client computer 72.

As shown in FIG. 6, when the client computer 72 that initiated the connection receives the challenge 90, it
 15 decrypts the challenge 90 using the user's private key PRVKu and the policy agent's public key PUBKp. If the public key PUBKu used to encrypt the challenge 90 indeed belongs to the user, the client computer would be able to decrypt the challenge to obtain the hash number H1. The client computer
 20 then constructs a response 94 that ties the user authentication to the network data stream. To prepare the response 94, the client computer concatenates a random number RND2, the decrypted hash H1, and data DATAnp from the first N packets of the transmitted network data, where N is a pre-
 25 selected number, such as five (5). The concatenated value 150 is then used as the input for a hash function 152, such as the

65601" 5579750

MD5 algorithm, to generate a second hash value H2. This step is represented by the following expression:

$$H2 = MD5(RND2 * H1 * DATA_{np}).$$

The hash value H2 is then concatenated with the random number RDN2, and the concatenated value 154 is encrypted first with the user's private key PRVKu and then with the public key PUBKp of the policy agent to form the response 94. This step is represented by the following expression:

$$R = PUBKp(PRVKu(H2 * RND2)),$$

where R is the response. The response 94 is then sent to the policy agent 78.

Turning now to FIG. 7, when the policy agent 78 receives the response 94 from the client computer, it decrypts the response using its own private key PRVKp and the public key PUBKu of the user to obtain the hash H2 and the random number RDN2. To verify that the hash number H2 from the decrypted response is what it should be, the policy agent calculates a hash value H2p using the hash value H1 included in the challenge 90, the random number RND2 from the decrypted response 94, and the data RDATA_{np} of the first N packets of the received network data as the input for the hash function.

An agreement between the hash value H2p calculated by the policy agent and the hash value H2 decrypted from the response 94 indicates that the user is who she claims to be, and that the network data received by the policy agent from the client computer are indeed those sent by the user. In other words, the validation of the response serves two functions:

authenticating the user and associating the received network with the authenticated user. The user is authenticated because only the real user would have the private key to properly decrypt the challenge and construct a valid response.

- 5 The network data are confirmed to be those sent by the user because if the data packets received by the policy agent have been maliciously altered, the hash value H2p calculated by the policy agent would be different from the hash value H2 obtained from the response. In addition to the user
- 10 authentication and data stream association, the hash value H2 can now be used as a one-time shared secret for further session encryption.

- In the embodiment described above, the public keys PUBKp and PUBKu of the policy agent and the user, respectively, are
- 15 used for encryption in the challenge-response process. In another embodiment, to speed up the process, the policy agent can opt not encrypt the challenge with the public key PUBKu of the user, and the client computer will also not encrypt the response with the public key PUBKp of the policy agent. This
- 20 process saves two public-key encryption steps and the corresponding private-key decryption steps. In this case, however, the hash value H2 may be obtained by a malicious attacker by intercepting the response and decrypting it with the public key PUBKu of the user. As a result, the hash H2 is
- 25 preferably not used as a shared secret for further encryption.

In view of the foregoing, it can be seen that the present invention provides an effective method and system for a policy

00436135-10909
SECRET

5

10

15

20